Machine Learning for the Prediction of App Energy Consumption from Appstore Data

Daniel Afonso Valente - M11060

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática
(2º ciclo de estudos)

Orientador: Prof. Doutor Luis A. Alexandre

Covilhã, Outubro 2022.

Declaração de Integridade

Eu, Daniel Afonso Valente, que abaixo assino, estudante com o número de inscrição M11060 de/o Engenharia Informática da Faculdade de Engenharias, declaro ter desenvolvido o presente trabalho e elaborado o presente texto em total consonância com o **Código de Integridades da Universidade da Beira Interior**.

Mais concretamente afirmo não ter incorrido em qualquer das variedades de Fraude Académica, e que aqui declaro conhecer, que em particular atendi à exigida referenciação de frases, extratos, imagens e outras formas de trabalho intelectual, e assumindo assim na íntegra as responsabilidades da autoria.

Universidade da Beira Interior, Covilhã 10 /10 /2022

Daniel Afonso Valente

(assinatura conforme Cartão de Cidadão ou preferencialmente
assinatura digital no documento original se naquele mesmo formato)

Acknowledgements

A deep thank you to my advisor, professor Luis Alexandre, for all his patience, availability and tremendous help during the whole project. A thank you to Phd student Nuno Pereira, for all the support and assistance that he provided throughout the project. A thank you to all my friends and family who have helped me carry on and a thank you to all the staff of the university for all your hard work and the amazing years that this university has provided me. This work was financed by FEDER (Fundo Europeu de Desenvolvimento Regional), from the European Union through CENTRO 2020 (Programa Operacional Regional do Centro), under project CENTRO-01-0247-FEDER-047256 – GreenStamp: Mobile Energy Efficiency Services.

Resumo

O mercado dos dispositivos móveis tem visto um tremendo desenvolvimento nos últimos anos tanto em termos de *hardware* como de *software* que é disponibilizado para os dispositivos. Apesar disto, as baterias que abastecem estes dispositivos não têm tido melhorias e têm sido incapazes de acompanhar o progresso desta área.
Devido a este fenómeno, os investigadores têm vindo a mostrar um interesse cada vez maior no ramo de soluções para computação verde de modo a gastar o mínimo de energia possível com dispositivos móveis. Isto gerou uma variedade de respostas, desde determinar o consumo energético de uma aplicação de forma acertada com recurso a modelos e *profilers* energéticos até ao desenvolvimento de práticas de codificação adequadas para a conservação da energia dos dispositivos.
No entanto, têm havido poucos estudos realizados sobre soluções destinadas aos utilizadores que ajudem a resolver este problema. De modo a preencher esta lacuna, este estudo foca-se no desenvolvimento de uma solução de aprendizagem automática que determine as conexões entre a informação sobre uma aplicação na *appstore* e o seu consumo energético. Sendo assim, o principal contributo deste projeto reside na aprendizagem automática mencionada anteriormente, adaptada para a *appstore* Aptoide e com principal foco nas aplicações pertencentes à categoria de jogos sendo que estas compõem grande parte do volume de descarregamentos da plataforma.

Palavras-chave

computação verde, aprendizagem automática, móvel, energia, Aptoide, consumo energético

Resumo alargado

O mercado dos dispositivos móveis tem visto um tremendo desenvolvimento nos últimos anos tanto em termos de *hardware* como de *software* que é disponibilizado para os dispositivos. Apesar disto, as baterias que abastecem estes dispositivos não têm tido melhorias e têm sido incapazes de acompanhar o progresso desta área.
Devido a este fenómeno, os investigadores têm vindo a mostrar um interesse cada vez maior no ramo de soluções para computação verde de modo a gastar o mínimo de energia possível com dispositivos móveis. Isto gerou uma variedade de respostas, desde determinar o consumo energético de uma aplicação de forma acertada com recurso a modelos e *profilers* energéticos até ao desenvolvimento de práticas de codificação adequadas para a conservação da energia dos dispositivos.
No entanto, têm havido poucos estudos realizados sobre soluções destinadas aos utilizadores que ajudem a resolver este problema. De modo a preencher esta lacuna, este estudo foca-se no desenvolvimento de uma solução de aprendizagem automática que determine as conexões entre a informação sobre uma aplicação na *appstore* e o seu consumo energético. Sendo assim, o principal contributo deste projeto reside na aprendizagem automática mencionada anteriormente, adaptada para a *appstore* Aptoide e com principal foco nas aplicações pertencentes à categoria de jogos sendo que estas compõem grande parte do volume de descarregamentos da plataforma.
Este projeto é composto por uma breve introdução ao tema assim como motivações para o desenvolvimento do mesmo, seguido por uma análise do trabalho relacionado, em seguida é descrito o processo de recolha de dados necessários para a execução da experiência. No total, 57 aplicações foram analisadas, com recurso a um dispositivo de medição improvisado, e foram utilizadas num conjunto de dados final contendo as permissões que cada aplicação necessita, o seu tamanho em memória, a versão mínima do android e o número total de permissões que tem. Este capítulo é procedido por uma descrição e análise dos resultados, todas as experiências foram feitas de modo a possibilitar que sejam replicadas e foram utilizados 4 modelos no total, estes são a regressão linear, regressão de vetores de suporte, Naive Bayes e *K-Nearest Neighbor*. Desta análise percebe-se que o modelo de regressão linear tende a fazer sobre-ajuste do conjunto de dados desenvolvido, sendo que os resultados mais promissores pertencem ao modelo de regressão de suporte de vetores com uso de um kernel de função de base radial. O documento termina com uma breve conclusão do trabalho realizado, denotando que parte dos problemas se devem ao tamanho reduzido do conjunto de dados mas que, mesmo assim, os resultados indicam boas oportunidades para a aplicação de aprendizagem automática na previsão de consumos energéticos das aplicações através da informação que têm disponível na *appstore*.

Abstract

The mobile market has seen tremendous development throughout the past few years both in terms of hardware and the software that is available for the devices. Despite this, the batteries that power these devices have not seen major improvements and have been unable to accompany the progress seen in this field. Due to this phenomenon, researchers have been showing a growing interest in the development of green computing solutions in order to spend the least amount of energy possible when using mobile devices. This as presented itself in a plethora of ways, from the accurate evaluation of the energy consumption of applications through the use of energy models and profilers to the assessment and development of better coding practices with energy conservation as the main focus. However, there have been few to no studies regarding the development of user-side solutions to help solve this problem. In order to fill this gap in research this study focuses on providing a machine learning solution with the intent of identifying links between the information available in the store page of an application and its energy consumption to develop an *a priori* method for the classification and certification of mobile applications. Hence the main contribution of this project resides on the previously mentioned machine learning model, adapted to the Aptoide appstore and mainly targeting applications that belong to the games category, given that these have the highest volume of downloads and interest by the users of the appstore.

Keywords

green computing, machine learning, mobile, energy, Aptoide, energy consumption

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

APP	Application
APK	Android Aplication Pack
ARO	Application Resource Optimizer
CNN-FT	Convolutional Neural Networks-Fault Tolerance
CPU	Central Process Unit
CSV	Comma Separated Value
DuT	Device under Test
MPSOCD	Multi-objective Particle Swarm Optimization Crowding Distance
SEMO	Smart Energy Monitoring System
UBI	Universidade da Beira Interior
UI	User Interface
SVR	Support Vector Regression
Poly	Polynomial
RBF	Radian Basis Function
MSE	Mean Square Error

Chapter 1

Introduction

1.1 Introduction

One of the major concerns nowadays is the energetic consumption of the various instruments and tools utilised for the various tasks. Mobile devices are no exception and this consumption is ruled by not only the time spent active on them but also which apps are being used and what activities they execute on background or when idled by the user. Studies have shown that a smartphone takes up from 3.5 to 4.9 kWh a year as its power demand [Alm21]. For reference purposes, Portugal has an estimated consumption of 4.5 kWh *per capita* annually[Wor21] meaning that a single mobile device consumes almost as much energy as a person living in Portugal. Not only this, but mobile devices also face a restriction that few appliances do, this is the need for a battery, which limits the effective usage time of the device and may be longer or shorter according to the usage given to the device. Also, the development of batteries has been unable to keep up with the rapid increase in capability of processors, hard disks and memory which in turn requires more and more power in their various uses.
For these reasons, it is understandable why the energy consumption of mobile devices is of such great importance for the public and why as of recent years the focus on green computing has been steadily rising in order to find different means to save energy, this may come in the form of adequate code practices, energy profilers, energy rating schemes or sets of advised practices so the user can better manage the energy consumption of their device.

1.2 Motivation

Despite the many contributions existing in the field of green computing to this problem, it is yet to be developed and implemented a concise and trustworthy energy consumption rating system for mobile applications, furthermore have it done in a manner that does not increase by much, almost none if possible, the amount of time it takes for the deployment of the application in the various app stores.
It is with this in mind that the author is motivated to investigate a solution to this problem, since many benefits can arise from this both for the user and appstore perspectives, since that, the user will acquire more information about the application and be better informed about the application he is downloading and its effects on the devices' battery and an increased interest by the user may arise in continuing to use that appstore since it allows him to make better decisions.

However to achieve this solution it is necessary the development of tools and methods that can acquire the needed information to be processed and result in an accurate and trustworthy energy rating of the app.

1.3 Thesis Focus and Scope

The main focus of this thesis lies in the development of a machine learning solution that, through the information provided in any given app Aptoide's page, can extract, in an efficient way, features and associate them with the energy consumption of the respective app so as to provide an adequate result that will later be utilized in the process of developing an accurate energy rating for the aforementioned app.
Despite the fact that the solution must work for any given app, the category of apps chosen to be the subject of this thesis is games. This is due to Aptoide's main focus being on this category given that it is the one that most of its users prefer as well as the fact that the most complex and energy draining applications also reside in this category since they make the most use of all the components that exist in a mobile device and often times to their full extent [CPJ+21], allowing for an overall analysis of the impact in terms of energy consumption of each component.
In order to achieve this main goal a subset of objectives are required to be accomplished first, these are:

1. In-depth research of the field of green computing pertaining to mobile devices in order to better understand available methods and solutions for energy consumption measurement and any pitfalls that these may incur when providing this service. Research also needs to be conducted on machine learning models in order to pick the most adequate.

2. Develop a database containing a set of the most downloaded applications that fall under the category mentioned previously together with all the pertinent information provided in their Aptoide page and respective measurement of the energy consumption of these apps with the use of the varied methods and solutions acquired through the performed research;

After these steps are completed, it will then be possible to develop the model mentioned in the first paragraph of this section which will then be tested in accordance with the purpose it will perform later.

1.4 Document Structure

This document is structured in the following manner:

1. **First Chapter**(Introduction): The first chapter is comprised of an introduction to the main theme in which the project falls under and its objectives, ending with the structure that this document presents;

2. **Second Chapter**(Related Work): The related work contains all the research gathered so far and found to be of major importance to the work being developed in this thesis. It contains in three subsections found to be the main points of focus of this project which are:

   (a) **Research**: This section focuses on the varied articles pertaining the research developed until now in the field of green computing with the main focus on energy rating models and analysis of batteries and the behavior of energy in mobile devices;

   (b) **Existing Energy Profilers**: Discusses the currently available energy profilers and surveys made on them can be found in this section;

   (c) **Machine Learning Algorithms**: Presents articles that implement a machine learning solution related to green computing and mobile devices and a brief analysis of the available models that can be of use for this project are presented in this section.

   The chapter ends with a small conclusion on all the insights gathered from this research;

3. **Third Chapter**(Experiment Setup, Models and Concepts): This chapter focuses on establishing all the methods that were used and all the work that was performed necessary before the testing phase;

4. **Fourth Chapter**(Result Analysis and Threats to Validity):A comprehensive analysis of the gathered results is performed in this chapter along with a description of the testing conditions to allow repeatability and possibles threat to the validity of the work performed;

5. **Fifth Chapter**(Final Conclusions and Future Work):The last chapter contains a structured understanding of the work yet to be done and a conclusion on the insights gathered with this experiment.

Chapter 2

Related Work

2.1 Introduction

This chapter presents the current state of technology regarding green computing applied to mobile devices. The research available is presented and discussed in three different sections, each focusing on a key aspect of this thesis that will later prove itself useful in the development of the solution in which the thesis is focused on.

The Research section presents articles that provide insight into energy consumption analysis and its importance for the user.

The section Existing Energy Profilers will present the energy profilers for mobile devices which are software and hardware tools available to attain predictive models for the energy consumption of mobile devices and an extensive look into which ones may suit the intended use for this project.

Lastly, the Machine Learning Algorithms will present insights into the available models to solve the problem that is the main focus of this research.

All research presented in this chapter was developed in the last five years with the exception of two articles [HSK+15] [AGH+15] presented in the Existing Energy Profilers, were conducted seven years ago however they contain an extensive list of profilers and in depth analysis of said profilers results that was deemed important for a better understanding of this subject matter.

2.2 Research

2.2.1 Power Consumption Analysis, Measurement, Management, and Issues: A State-of-the-Art Review of Smartphone Battery and Energy Usage

The article [PSM+19] provides an in-depth look at how smartphones and their batteries function and how to measure and reduce the energy consumption through the use varied techniques. It also analyses the available research into the development of smartphone batteries and hazards associated with these, providing some solutions on how to mitigate them.

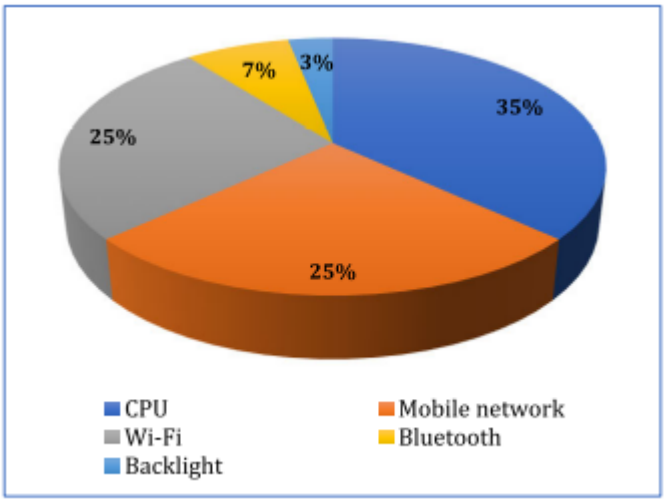

Figure 2.1: Power consumption percentage of different components in a smartphone [PSM+19].

Despite the main focus of the article being the smartphone batteries, its contributions to this thesis come mainly from its comprehensive study into the energy consumption of the various components 2.1 of the smartphone from which the author attains a surface understanding of the energy consumption distribution percentiles between them and of the available profilers and models for measurement and diagnosis of energy consumption 2.2 providing further information on the development of these software tools.

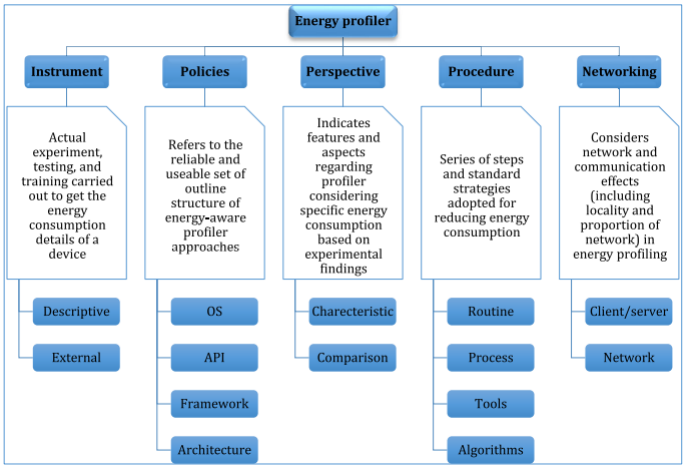

Figure 2.2: Key elements of an energy profiler [PSM+19].

2.2.2 Google Play Apps ERM: (Energy Rating Model) Multi-Criteria Evaluation Model to Generate Tentative Energy Ratings for Google Play Store Apps

The thesis [Alm21] was conducted with the intent to develop a mixed strategy between user side preventive power-saving plans with conventional detective strategies and thus come up with an adequate rating system for the apps present in the Goggle Play Store.

Through an in depth look at the current state of green computing, user experience and the information made available by Google in their app store, more specifically the list of permissions pertaining to the app, the author was able to develop a energy rating scheme with the use of stars to represent the quality in terms of energy usage of the application. This thesis utilizes Power Tutor in order to develop an energy model of the Samsung I9500 so as to then apply the estimated consumption of the varied components and then associate them to their respective permissions as presented in 2.1.

Average Energy Consumption Amount and Ratings of Energy Consuming Permissions Table			
Power Consuming Applications Permissions	Amount of Energy Consumption of each Used Component	Permission Average Energy Consumption Amount per minute	Permission Star Rating out of Six Stars (∼1 to ∼30 mAh)
Access Bluetooth settings	Bluetooth Radio (∼10 mAh)	∼10mAh	⋆⋆
Allow Wi-Fi Multicast reception	Wi-Fi Radio (∼12mAh)	∼12 mAh	⋆⋆
Broadcast data message to apps	Wi-Fi Radio(∼12mAh) Cellular Radio(∼17 mAh)	∼15 mAh	⋆⋆⋆
Change network connectivity	Wi-Fi Radio(∼12mAh) Cellular Radio(∼17 mAh)	∼15 mAh	⋆⋆⋆
Change system display settings	Screen(∼16 mAh)	∼16 mAh	⋆⋆⋆
Change your audio settings	Audio Speaker (∼9 mAh)	∼9 mAh	⋆⋆
Change/intercept network settings and traffic	Wi-Fi Radio(∼12mAh) Cellular Radio(∼17 mAh) Application Processor (∼20mAh)	∼16 mAh	⋆⋆⋆
Connect and disconnect from Wi-Fi	Wi-Fi Radio (∼12mAh)	∼12 mAh	⋆⋆
Control flashlight	Flash Light (∼19 mAh)	∼19 mAh	⋆⋆⋆⋆
Control vibration	Vibration (∼15 mAh)	∼15 mAh	⋆⋆⋆

Continuation of 2.1			
Power Consuming Applications Permissions	Amount of Energy Consumption of each Used Component	Permission Average Energy Consumption Amount per minute	Permission Star Rating out of Six Stars (∼1 to ∼30mAh)
Directly call phone numbers	Cellular Radio (∼17 mAh) Microphone (∼5 mAh) Audio Speaker (∼9 mAh)	∼10 mAh	⋆⋆
Download files without notification	Wi-Fi Radio (∼12 mAh) Cellular Radio (∼17 mAh) Application Processor (∼20 mAh)	∼16 mAh	⋆⋆⋆
Full network access	Wi-Fi Radio (∼12 mAh) Cellular Radio (∼17 mAh) Application Processor (∼20 mAh)	∼16 mAh	⋆⋆⋆
Make an app to always run	Application Processor (∼20 mAh)	∼20 mAh	⋆⋆⋆⋆
Modify phone state	Application Processor (∼20 mAh)	∼20 mAh	⋆⋆⋆⋆
Modify secure system settings	Application Processor (∼20 mAh)	∼20 mAh	⋆⋆⋆⋆
Modify system settings	Application Processor (∼20 mAh)	∼20 mAh	⋆⋆⋆⋆
Pair with Bluetooth devices	Bluetooth Radio (∼10 mAh)	∼10 mAh	⋆⋆⋆⋆
Precise (GPS) location	GPS (∼25 mAh)	∼25 mAh	⋆⋆⋆⋆⋆

Continuation of 2.1			
Power Consuming Applications Permissions	Amount of Energy Consumption of each Used Component	Permission Average Energy Consumption Amount per minute	Permission Star Rating out of Six Stars (∼1 to ∼30mAh)
Prevent phone from sleeping	Application Processor (∼20 mAh) Screen (∼16 mAh)	∼18 mAh	⋆⋆⋆⋆
Read your social stream	Wi-Fi Radio (∼12 mAh) Cellular Radio (∼17 mAh) Application Processor (∼20 mAh)	∼16 mAh	⋆⋆⋆
Record audio	Microphone (∼5 mAh)	∼5 mAh	⋆
Run at startup	Application Processor (∼20 mAh)	∼20 mAh	⋆⋆⋆⋆
Send sticky broadcast	Wi-Fi Radio (∼12 mAh) Cellular Radio (∼17 mAh) Application Processor(∼20 mAh)	∼16 mAh	⋆⋆⋆
Take pictures and videos	Cameras (∼17 mAh) Flash Light (∼19 mAh) Microphone (∼5 mAh)	∼14 mAh	⋆⋆⋆
Toggle sync on and off	Wi-Fi Radio (∼12 mAh) Cellular Radio (∼17 mAh) Application Processor (∼20 mAh)	∼16 mAh	⋆⋆⋆
View Wi-Fi connections	Wi-Fi Radio (∼12 mAh)	∼12 mAh	⋆⋆

Continuation of 2.1			
Power Consuming Applications Permissions	Amount of Energy Consumption of each Used Component	Permission Average Energy Consumption Amount per minute	Permission Star Rating out of Six Stars (∼1 to ∼30mAh)
Write to your social stream	Wi-Fi Radio (∼12 mAh) Cellular Radio (∼17 mAh) Application Processor (∼20 mAh)	∼16 mAh	⋆⋆⋆

Table 2.1: Average power consumption table and ratings for power consuming permissions as per [Alm21].

From the excerpt of this table we can see that the authors of the study managed to produce a relatively, although simplistic, rating for the energy consumption of each permission through the mobile components they utilize. All of this was possible due to the correlation between the data attained from Power Tutor and human analysis.
Given the subject matter of this paper, this study is of high relevance, allowing the author to attain an overview of the recent state of green computing with focus on the mobile market and to gather valuable insights on how to approach the problem meant to be solved by this thesis.

2.3 Existing Energy Profilers

2.3.1 Modeling, Profiling, and Debugging the Energy Consumption of Mobile Devices

The focus of the article [HSK+15] lies in an extensive explanation on how to develop an energy profiler and the analysis of the literature related to several energy profilers with the intent of comparing their performance and ease of use. The authors divide the profilers into three different categories, the categories and respective profilers are the following:

1. On-Device profiler with on Device model:
    - Nokia Energy Profiler;
    - Trepn Profiler;
    - Power Booter;
    - Se-same;
    - DevScope;
    - AppScope;
    - V-edge.
2. On-Device Profiler with off-Device Model:
    - Android Power Profiler;
    - Power Tutor;
    - Power Prof.
3. Off-device in Laboratory:
    - PowerScope;
    - Joule Watcher;
    - Fine-grained Profiling with Eprof;
    - Banerjee et al;
    - Shye et al.

Profiler	Display	CPU	GPU	GPS	BT	Wi-Fi	3G	4G	Cam.	SD Card	Audio	Reported Accuracy
Trepn	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	99%
V-edge	✓	✓	×	✓	×	✓	×	×	×	×	×	86%
BatteryStats	✓	✓	×	✓	✓	✓	×	×	×	×	×	Not Reported
PowerBooter	✓	✓	×	✓	×	✓	✓	×	×	×	✓	96%
PowerTutor	✓	✓	×	✓	×	✓	✓	×	×	×	✓	97.5%
DevScope	✓	✓	×	✓	×	✓	✓	×	×	×	×	95%
AppScope	✓	✓	×	✓	×	✓	✓	×	×	×	×	92%
Sesame	✓	✓	×	×	×	✓	×	×	×	✓	×	86%
Eprof	✓	✓	×	✓	×	✓	✓	×	✓	✓	×	94%
Banerjee et al.[2014]	✓	✓	×	✓	×	✓	✓	×	×	×	✓	Not Reported
Shye et al. [2009]	✓	✓	×	×	×	✓	×	×	×	✓	×	93%

Table 2.2: Table that presents various energy profilers together with their accuracy and components analyzed as per [HSK+15]

The main contribution of this article is the pooling of available profilers, their respective accuracy values and comparison of components between them allowing for an easier choice of which profiler to apply to this thesis. From 2.2 it is possible to determine that the Trepn profiler has the best accuracy and analyses the consumption of most components that exist in smartphones, however these accuracy values are those reported by the developers of the various profilers and were not tested by the authors of this article and thus, the values may be biased towards the applications that were used by the developers to test the profilers.

2.3.2 A Review on mobile application energy profiling: Taxonomy, state-of-the-art, and open research issues

The main intent of this study [AGH+15] is the development of a concise taxonomy so as to facilitate the understanding and separation of the varied existing methods available for energy profiling of mobile devices, this is then substantiated through an analysis of different methods that suit the two main categories the authors theorized, these categories and profilers are the following:

1. Software based:
    - Power-prof;
    - Power Booter;
    - Se-same;
    - Hybrid-feedback;
    - SEMO;
    - Elens;
    - ARO;

- Wattson;
- Eprof;
- P-top.

2. Hardware based:

   - DuT;
   - Netw-trace;
   - Power Memo;
   - PowerScope;
   - Network;
   - Web-browser;
   - Multi-core CPU.

For the purposes of this article, the authors focused mainly on the analysis of the software energy profilers as seen in 2.3.

Schemes (Ref.)	Resources analysed								Description
	CPU	LCD	GPS	Wi-Fi	UMTS	GSM	Accelerometer	Compass	
Power Proff (Kjærgaard and Blunck, 2012)	✓		✓	✓		✓	✓	✓	Unsupervised GA based energy profiling
Power Booter (Zhang et al., 2010)	✓	✓	✓	✓	✓				Automated smart battery interface based power model construction
Se-same (Dong and Zhong, 2010)	✓	✓		✓					High rate and accurate self-power modeling
Hybrid-feedback (Gurun and Krintz, 2006)	✓					✓			Hybrid (on-line/off-line) feedback based energy profiling to reduce power model construction time
SEMO (Ding et al., 2011)	✓								Low-rate smart energy monitoring system based profiling to improve accuracy
ELens (Hao et al., 2013)	✓		✓	✓					Program analysis based fine-grained energy profiling
ARO (Qian et al., 2011)	✓				✓				Cross-layer interaction based power optimization
Wattson (Mittal et al., 2012)	✓	✓		✓	✓				Emulation based profiling
Eprof (Pathak et al., 2012)	✓		✓	✓					Fine-grained energy profiling for smart phone applications
P-top (Do et al., 2009)	✓	✓				✓			Process level software power profiling

Table 2.3: Table that presents several profilers and their descriptions as well as a list of what components each analyses as per [AGH+15]

This article contributes to a deeper and concise knowledge on the available energy profilers and their capabilities. Its contribution for this paper lies on widening the pool of available profilers.

2.3.3 GreenScaler: training software energy models with automatic test generation

GreenScaler [CH18] is a robust energy model developed with Android developers in mind so as to allow them to estimate the energetic consumption of the applications that they develop. This energy model software makes use of random test generation and a CPU usage focused heuristic in order to select which test cases are best, through this, GreenScaler is able to produce relatively accurate energy models for an application with an upper error

bound of 10% when using randomly generated test cases and an even lower error percentage when applied to manually written tests. It is also able to detect energy regression within different versions of an app as long as it's a relevant difference.

This energy model software is able to achieve its purpose, through the use of GreenMonkey, an adaptation of Android Monkey the UI/Application exercisers. This adaptation was developed since the original was somewhat limited, mainly because it didn't allow for the user to define an event distribution and it included irrelevant events as well that aren't related to the app itself. Later in the research the authors compared their version's performance to the original's and found that their version produced much better results when the CPU-utilization heuristics were applied than the original and thus opted with GreenMonkey for automatic test generation.

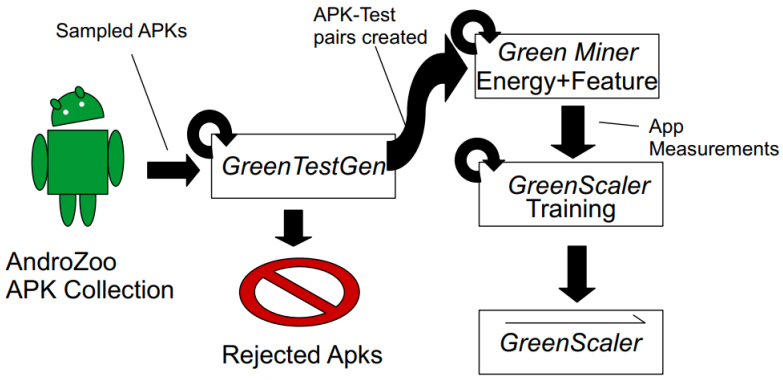

Figure 2.3: GreenScaler Architecture as per [CH18]

The authors evaluation of both resource usage heuristics, these being a CPU-utilization heuristic and an energy estimation heuristic, was done by using the same feature table but using the different datasets obtained from implementing the different heuristics to the test selection process. The leave-one-out method was applied so as to test the accuracy of the model, the Anderson-darling normality test was also utilized and it determined that both error distributions are not normally distributed and thus the Kruskal-Wallis test was used which yielded that the error distributions from the models obtained out of the different heuristics were statistically different. Due to this, the Wilcoxon-rank-sum test was applied to attain the 99% confidence interval of mean percent error in joules as well as the Cliff's delta to measure the effect size between them.

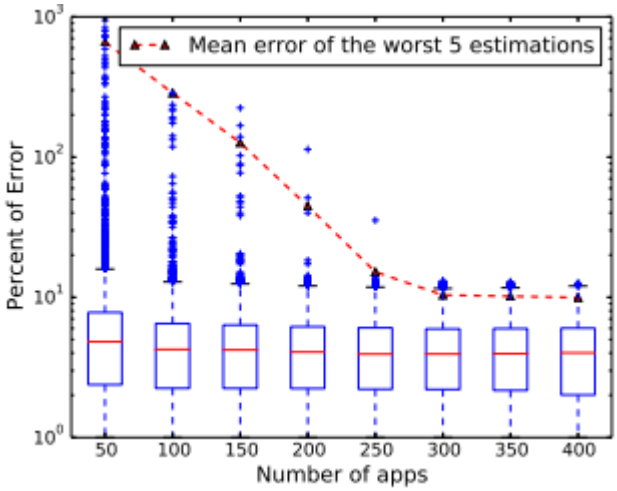

Figure 2.4: Accuracy table for GreenScaler [CH18].

After the statistical analysis was complete, the chosen model for GreenScaler was the Model based off of CPU-utilization, due to it having a better upper error bound and a difference of 3% in the mean error for the 5% worst estimations. The CPU heuristics model also benefits from it's simplicity since it only requires to capture the CPU jiffy (period of an alternating current power cycle) information whereas the energy model heuristic requires far more data and to trace every single system call by an app. The research detailed in this report is highly relevant due to important insights gathered from it that will benefit the projects future development, e.g. when implementing a test selection, a simple CPU-utilization heuristic seemed to be the best performing one, being preferred over the use of a more complex energy estimation or a code coverage heuristic. In terms of training the model it was found that with 400 different apps in the training set, it's able to reach the upper error-bound of 10%, however the error rate decay slows down considerably after using 300 apps in the training set as seen in 2.4.

2.3.4 Software-Based Energy Profiling of Android Apps: Simple, Efficient and Reliable?

PETrA [DNPP+17] or Power Estimation Tool for Android is a software-based tool developed with the intent of understanding if it was possible to build such a tool and attain accurate energy consumption measurements without the use of hardware-based tools that are expensive to acquire but provide the most accurate measurements.

This tool functions in the following manner, firstly it starts by handling app preprocessing in which PETrA is given an app, it's directory and is then tasked with installing and preparing the app for analysis, then it proceeds to compute the energy profile of the app by receiving a test case, this test can be a manually written one or one generated automatically.

After the test is finished then, through the use of Project Volta's Android tools, PETrA is capable of creating a power profile of the app and from it, together with the use of certain

formulas, pinpoint how much energy the app consumes when performing certain tasks, finally the tool generates an output, a csv file with the energy estimations for each of the method calls the app executes.

2.3.5 EMaaS: Energy Measurements as a Service for Mobile Applications

The paper [CA19] details the development of a peer-to-peer cloud based system with the intent to supply mobile developers with a platform in which they can have access to various energy models and hardware analysis methods so they can assess the energy consumption levels of the applications they are developing. This is an incredibly useful feature since it allows developers to analyze the energy consumption of their application without having to acquire different sets of tools and knowledge which they may not have and may be expensive to acquire by themselves.

In order to accomplish this feat, the authors developed the platform with three different types of users in mind, being these the Developers, users with the intent of using the platform for the assessment of the energy consumption of the various apps they develop, the Providers, users who supply the network with devices which will allow the applications from developers to run on said devices in order to estimate the energy consumption resorting to adequate energy models to achieve this end, and finally the Super-Providers, who will provide hardware based energy measurements in order to create a trust value for comparison with the energy model acquired measurements.

Despite the three different types of user it is possible for any one user to meet requirements for all three types and fulfill all tasks simultaneously.

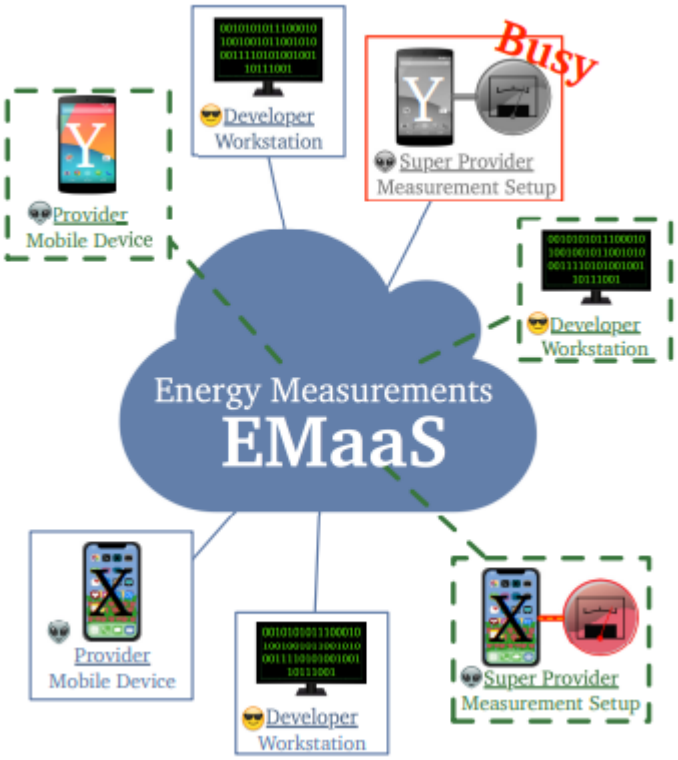

Figure 2.5: EMaaS architecture [CA19].

The process of measuring said consumption occurs has follows:

1. Developer requests assessment for his application by providing an APK to the platform along with an instrumentation build and test cases for said app;
2. APK is sent to a Provider and a Super-Provider, for this to be properly processed it is required that both these parties have the same device model as the Developer;
3. Provider returns energy consumption measurements based off of a energy model adequately arranged for the device in question whilst the Super-Provider will provide measurements based off of readings from the monitoring of hardware readings with the use of proper equipment for the task;
4. Results are returned to the Developer and the process is concluded.

This process sustains itself on two major implemented mechanisms to assess which results can be delivered to the Developer so he can attain accurate data, these are the Reliability Consultant, the Energy Model available, the Hardware-based power Monitor is also crucial however it's only used has a trust base system for the previous two instead of being an implemented algorithm.
When a Developer requests an energy measurement this request will first go through the reliability consultant in order to determine if the available energy model for said device model is enough to accurately determine the measurement on it's own. It does this through applying the values obtained from both the Provider and the Super-Provider in a

mathematical formula that will evaluate said energy model's reliability, the closer to zero the result is the more reliable the model will be. In case the reliability of the model exceeds a certain threshold above or below zero, it will be considered as unreliable for the respective case and the values attained from the Super-Provider will be returned to the developer as well as they will also be used to update the existing energy model and the reliability consultant so that these are better prepared for future requests, otherwise the energy model attained value is used since it's assessed to be reliable.

2.3.6 PowDroid: Energy Profiling of Android Applications

Powdroid [BGN21] is a fusion between utilization-based and event-based energy profiler that checks the battery status at the moment of app initialization and compares it with its final value at the end of its execution. This is accomplished through the use of a Wi-Fi connection and a command-line tool with the phone and four specific tools, these are Batterystats, a tool in the Android framework that collects raw data from the battery, Bugreport, also from the Android framework and produces a zip file containing a report of the app's usage, Battery Historian, a tool from Google used to create an easy to read web-based visualization of the report provided by Bugreport and lastly, the authors, run a few scripts in order to split, process and unite the metrics in order to return a CSV file containing a list of all events and corresponding usage data.

This profiler was tested with three different types of apps, these were web browsers, camera and weather applications, the authors state that their results are consistent with the results of recent comparative studies.

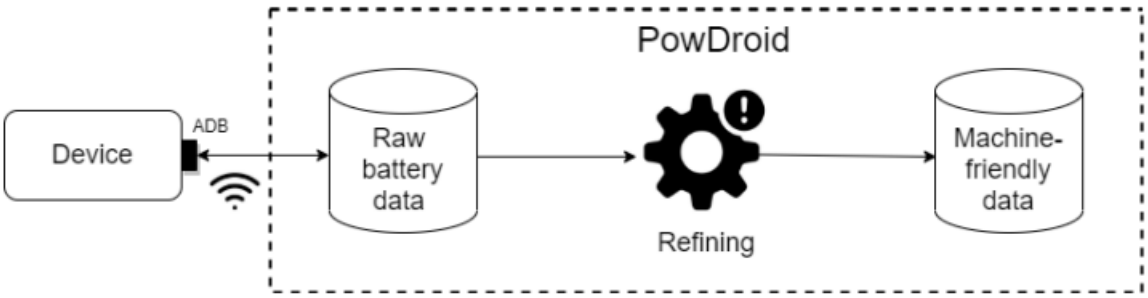

Figure 2.6: Powdroid architecture [BGN21].

The limitations of this profiler are the following: Monitors the phone battery consumption as a whole, relies on battery and hardware information provided by Batterystats hence, it is not in real time and not fine grained. Also, the values provided are estimations based on the metrics they applied which means that battery estimations are based on battery drain and the energy consumption of the individual components are not mapped.

2.4 Machine Learning Algorithms

This section is about ReviewViz: Assisting Developers Perform Empirical Study on Energy Consumption Related Reviews for Mobile Applications, given that it is the study, of all acquired, that has the most insights into the usage of machine learning pertaining the subject of this paper.
The study [HF20] deals with the development of a visualization tool to help developers better sift through the user reviews of their apps. This tool specifically deals with the topic of energy consumption and tests various machine learning models with the intent of choosing the best one for natural language assessment of the topic at hand.
The results from the study show that the CNN-FT, a type of Convolutional Neural Network with the added capability of fault tolerance, is the model that attains the best accuracy and F1-score values (approximately 0.94 for accuracy and 0.935 for F1-score as it is showed in 2.7) relinquishing however, the run-time, taking approximately 35 to 40 minutes to complete.

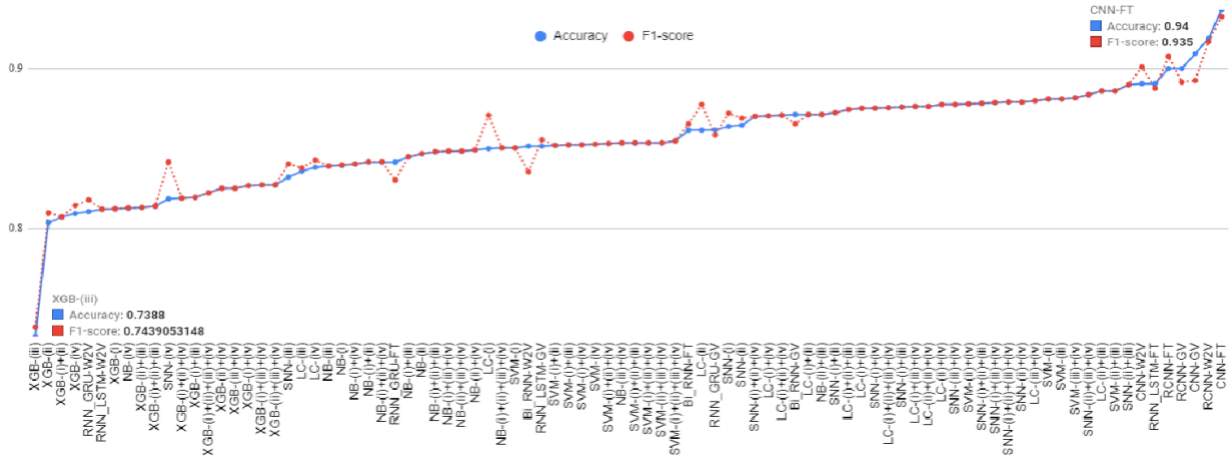

Figure 2.7: Accuracy and F1-Score for ReviewViz [HF20].

Although the purpose of this study is the development of a data visualization tool, it shows the possibility of using machine learning and user reviews to gather information about energy consumption, proving itself useful for the project being handled by the author of this thesis.

2.5 Conclusion

From this chapter it can be understood that, the research performed found no machine learning solution to study the correlation between energy consumption and the data readily available about an application in its appstore page. Despite this, all the tools and knowledge deemed necessary for the development of said solution exist and are publicly available.

Chapter 3

Experiment Setup, Concepts and Models

The efforts performed so far in order to further the progress of the thesis to reach its ultimate goal are discussed in this chapter and their relevance for the project highlighted and explained.

3.1 Dataset

A dataset has been developed comprised of eighty eight different applications found in the Aptoide app store within the category of games and across eighteen different subcategories. All these apps have been downloaded, at least, half a million times.Two different csv files have been created from the information available on each app with the intent of gathering the necessary features that will be supplied to the machine learning model for training and testing. The first dataset 3.1 is comprised of both categorical and numerical values, with the exception of the last two columns. These serve the purpose of ease of identification of the apps within the app store in case it is needed for future research. The second one 3.2 is comprised of a table relating each application to the respective permissions that they require. One hundred and twelve different permissions where identified in total across all the apps of the dataset.

App	Categoria	Tranferencias(	Tamanho (	NPermissoes	Min Android Ver.	Programador	Link
Fortnite: Battle Royale	Acao/Aventura/C	7.0	188.5	18	5.1	Epic Games	https://fortnite.pt.aptoide.com/app
Garena Free Fire	Acao	55.0	796.0	26	4.1	111dots Studi	https://freefire-111dots-studio.pt.aptoide.com/app
PUBG Mobile	Acao	38.0	1024.0	24	4.3	PROXIMA BET	https://pubg-mobile-tencent-games.pt.aptoide.com/app
Legacy of Discord	Acao	1.5	103.0	13	4.0.3	GTArcade	https://legacy.pt.aptoide.com/app
Mobile Legends	Acao	10.5	121.0	36	4.1	Moonton	https://mobile-legends.pt.aptoide.com/app
Call of Duty	Acao	11.5	232.0	20	4.3	Activision	https://com-activision-callofduty-shooter.pt.aptoide.com/app
Among Us	Acao	6.5	57.5	5	7.0	InnerSloth	https://among-us.pt.aptoide.com/app
Brawl Stars	Acao/Estrategia	13.0	140.0	14	4.3	Supercell	https://supercell-brawlstars.pt.aptoide.com/app
Clash Royale	Estrategia	56.0	133.5	13	4.1	Supercell	https://clash-royale.pt.aptoide.com/app
Clash of Clans	Estrategia/Arcad	54.0	121.0	9	4.4	Supercell	https://clash-of-clans.pt.aptoide.com/app
Clash of Kings	Estrategia	3.0	155.5	23	4.1	Elex Wireless	https://clash-of-kings.pt.aptoide.com/app
Lords Mobile	Estrategia	7.0	847.5	15	4.1	IGG.COM	https://lords-mobile.pt.aptoide.com/app
Roblox	Aventura	48.5	91.5	13	5.1	ROBLOX Corpc	https://roblox.pt.aptoide.com/app
Bed Wars	Aventura	0.804	137.5	16	4.1	Blockman Mul	https://bed-wars.pt.aptoide.com/app
Mini World	Aventura	5.5	358.5	32	4.4	SuperNice Digi	https://mini-world-miniplay-inc.pt.aptoide.com/app
Terraria	Aventura/Arcada	5.0	150.0	2	4.4	505 Games	https
Pokemon GO	Aventura	34.0	54.5	26	7.0	Niantic	https
60 Seconds_Atomic Adventure	Aventura	3.0	108.5	7	4.1	Robot Gentlen	https://60-seconds-robot-gentleman.pt.aptoide.com/app
Bendy and the Ink Machine	Aventura	4.5	534.5	4	7.0	Joey Drew Stu	https://bendy-joey-drew-studios.pt.aptoide.com/app
Perguntados(SP)	Cultura Geral/Qu	0.568	28.5	13	5.1	Etermax	https://trivia-crack-ad-free.pt.aptoide.com/app
Perguntados	Cultura Geral/Qu	2.5	43.0	15	5.1	Etermax	https://trivia-crack.pt.aptoide.com/app
Pokemon TCG Online	Cartas	0.633	346.5	4	4.1	The Pokemon	https://pokemon-trading-card-game-online.pt.aptoide.com/app
UNO	Cartas	0.6195	245.5	14	4.4	Mattel163 Lim	https://uno-mattel163-limited.pt.aptoide.com/app
Hearthstone	Cartas	732.0	1024.0	34	7.1	Blizzard Entert	https://hearthstone.pt.aptoide.com/app
Duel Links	Cartas	1.0	81.0	10	5.1	Konami	https://duel-links.pt.aptoide.com/app
Cut the Rope	Quebra Cabecas	2.5	51.0	8	4.2	ZeptoLab	https://cut-the-rope-free.pt.aptoide.com/app
Plague Inc	Quebra Cabecas	2.0	73.0	5	4.1	Miniclip	https://plague-inc.pt.aptoide.com/app

https://mini-world-miniplay-inc.pt.aptoide.com/app - Cliq
Clique e mantenha premido para selecionar esta célula.

Table 3.1: Structured dataset with categorical and numerical values.

Through the first dataset it will be possible to search for correlations between the size, number of permissions and minimum android version of the app and its total energy con-

sumption. The second dataset will be used to correlate the energy consumption to the various permissions of each app.

Permissoes									
App	Fortnite: Battle Royale	Garena Free Fire	PUBG Mobile	Legacy of Discord	Mobile Legends	Call of Duty	Among Us	Brawl Stars	Clash Royale
android.permission.ACCESS_NETWORK_STATE	1	1	1	1	1	1	1	1	1
android.permission.ACCESS_WIFI_STATE	1	1	1	1	1	1	0	1	1
android.permission.BLUETOOTH	1	1	1	0	1	1	0	0	0
android.permission.DOWNLOAD_WITHOUT_NOTIFICATION	1	0	0	0	0	0	0	0	0
android.permission.FOREGROUND_SERVICE	1	1	1	0	1	1	1	1	1
android.permission.GET_ACCOUNTS	1	1	0	0	1	0	0	0	0
android.permission.INTERNET	1	1	1	1	1	1	1	1	1
android.permission.MODIFY_AUDIO_SETTINGS	1	1	1	0	1	1	0	0	0
android.permission.RECEIVE_BOOT_COMPLETED	1	1	1	1	0	1	0	0	0
android.permission.RECORD_AUDIO	1	1	1	0	1	1	0	0	0
android.permission.REQUEST_INSTALL_PACKAGES	1	0	1	1	0	1	0	0	0
android.permission.VIBRATE	1	1	1	0	1	1	0	1	1
android.permission.WAKE_LOCK	1	1	1	1	1	1	0	1	1
com.android.vending.BILLING	1	1	1	0	1	1	1	1	1
com.android.vending.CHECK_LICENSE	1	1	1	0	0	0	0	0	0
com.app.permission.C2D_MESSAGE	1	1	1	0	1	1	0	0	0
com.google.android.c2dm.permission.RECEIVE	1	1	1	0	1	1	0	1	1

Table 3.2: Dataset comprised of permissions and apps.

From these two csv that comprised the dataset, only 57 apps were chosen to make the final dataset due to these being the only ones that could be tested in the test device used and were permitted to connect with their respective servers through the institution's Internet access.

This data was then filtered to remove any permission which had no app associated which meant these were empty features remaining 92 features of the initial 112, it's numerical values, were changed according to this equation:

$$\text{NewFeatVal} = \frac{\text{FeatVal}}{\text{MaxValFeat}}$$

	App	android.p	android.p	android.p	android.p	android.p	android.p	android.p	android.p	android.p	android.p	android.p	android.p	…	n.farsit	com.xiaor	android.p	android.p	android.p	com.andr	com.andr	com.andr	com.andr	com.andr	Tamanho	NPermissi	MInAndro	Media	Consumpt
6	Among Us	1	0	0	1	0	1	0	0	0	0	0	0	…	0	0	0	0	0	0	0	0	0	0	0.065564	0.147059	6	0.995714	1
10	Clash of K	1	1	0	0	0	1	1	1	1	0	1	1	…	0	0	0	0	0	0	0	0	0	0	0.177309	0.676471	2	1.839696	2
14	Mini Worl	1	1	1	1	0	1	1	0	1	0	1	1	…	0	0	0	0	0	0	0	0	0	0	0.40878	0.941176	4	1.21633	1
16	Pokemon	1	1	1	1	1	1	0	1	0	0	1	1	…	0	0	0	0	0	0	0	0	0	0	0.062144	0.764706	6	2.237704	0
17	60 Second	1	1	0	0	0	1	0	0	0	0	0	1	…	0	0	0	0	0	0	0	0	0	0	0.123717	0.205882	2	1.636834	2
18	Bendy and	1	0	0	0	0	1	0	0	0	0	0	0	…	0	0	0	0	0	0	0	0	0	0	0.609464	0.117647	6	2.329235	0
19	Perguntac	1	1	0	1	0	1	0	1	0	0	1	1	…	0	0	0	0	0	0	0	0	0	0	0.032497	0.382353	5	2.171554	2
20	Perguntac	1	1	0	1	0	1	0	1	0	0	1	1	…	0	0	0	0	0	0	0	0	0	0	0.049031	0.441176	5	2.171554	2
24	Duel Links	1	1	0	0	0	1	0	0	0	0	1	1	…	0	0	0	0	0	0	0	0	0	0	0.09236	0.294118	5	1.453003	1
25	Cut the Ro	1	1	0	0	0	1	0	0	0	0	0	0	…	0	0	0	0	0	0	0	0	0	0	0.083238	0.147059	2	2.752026	0
26	Plague Inc	1	0	0	1	0	1	0	0	0	0	0	1	…	0	0	0	0	0	0	0	0	0	0	0.210946	0.235294	4	2.56347	0
27	Robbery B	1	1	0	0	0	1	0	1	0	0	0	1	…	0	0	0	0	0	0	0	0	0	0	0.127138	0.647059	5	2.650513	0
28	Magic Tile	1	1	0	1	0	1	1	1	0	0	1	1	…	0	0	0	0	0	0	0	0	0	0	0.100342	0.323529	4	2.052127	2
29	Tiles Hop	1	1	0	0	0	1	0	1	0	0	1	1	…	0	0	0	0	0	0	0	0	0	0	0.082668	0.176471	5	2.444734	0
30	Incredibo:	1	0	1	0	0	1	0	0	0	0	0	0	…	0	0	0	0	0	0	0	0	0	0	0.036488	0.794118	4	2.342465	0
31	Solitaire	1	1	0	0	0	1	0	1	0	0	1	1	…	0	0	0	0	0	0	0	0	0	0	0.145382	0.529412	4	1.388551	1
32	FarmVille	1	1	0	1	0	1	0	1	0	0	1	1	…	0	0	0	0	0	0	0	0	0	0	0.09065	0.323529	5	1.253388	1
35	NBA LIVE	1	1	0	0	0	1	0	0	0	0	1	1	…	0	0	0	0	0	0	0	0	0	0	0.421323	0.411765	4	1.671898	2
36	Dream Lea	1	1	0	0	0	1	1	1	0	0	1	1	…	0	0	0	0	0	0	0	0	0	0	0.67959	0.588235	5	2.134799	2
37	eFootball	1	1	1	1	0	1	1	0	0	0	1	1	…	0	0	0	0	0	0	0	0	0	0	0.119156	0.294118	2	1.72961	2
40	FIFA Futel	1	1	0	0	0	1	0	0	0	0	1	0	…	0	0	0	0	0	0	0	0	0	0	0.079818	1	5	2.105149	2
41	True Skate	1	1	0	0	0	1	0	0	0	0	0	1	…	0	0	0	0	0	0	0	0	0	0	0.118586	0	0	2.095452	2
43	Avakin Lif	1	1	0	0	0	1	0	1	0	0	1	1	…	0	0	0	0	0	0	0	0	0	0	0.102052	0.705882	4	2.385864	0
44	FNAF Wor	0	0	0	0	0	0	0	0	0	0	0	0	…	0	0	0	0	0	0	0	0	0	0	0.644242	0.911765	4	2.516375	0
45	Marvel Fu	1	1	0	0	0	1	0	1	0	0	1	1	…	0	0	0	0	0	0	0	0	0	0	0.103193	0.323529	4	1.883117	2
46	Shadow Fi	1	1	0	1	0	1	0	1	0	0	1	1	…	0	0	0	0	0	0	0	0	0	0	0.033637	0.352941	5	2.094346	2
47	Star Wars:	1	1	0	1	0	1	0	0	0	0	1	0	…	0	0	0	0	0	0	0	0	0	0	0.110604	0.323529	4	1.662494	2
49	Pixel Art	1	1	0	0	0	1	0	0	0	0	1	1	…	0	0	0	0	0	0	0	0	0	0	0.082098	0.323529	4	1.754886	2
50	Bubble W	1	1	0	1	0	1	0	1	0	0	1	1	…	0	0	0	0	0	0	0	0	0	0	0.082098	0.323529	4	1.540589	2
51	Candy Cru	1	1	0	1	0	1	0	1	0	0	0	1	…	0	0	0	0	0	0	0	0	0	0	0.060433	0.205882	4	1.872116	2
52	Cut the Ro	1	1	0	0	0	1	0	0	0	0	0	0	…	0	0	0	0	0	0	0	0	0	0	0.118586	0.176471	1	1.44494	1
53	Gacha Life	1	0	0	1	0	1	0	0	0	0	0	0	…	0	0	0	0	0	0	0	0	0	0	0.098062	0.323529	4	1.754886	2
54	Candy Cru	1	1	0	1	0	1	0	1	0	0	0	1	…	0	0	0	0	0	0	0	0	0	0	0.08894	0.294118	4	2.160433	2
55	Meu Talki	1	1	0	0	0	1	0	1	1	0	0	1	…	0	0	0	0	0	0	0	0	0	0	0.103763	0.323529	4	1.754886	2
56	Candy Cru	1	1	0	1	0	1	0	1	0	0	0	1	…	0	0	0	0	0	0	0	0	0	0	0.103763	0.323529	4	1.648539	2
57	Farm Hero	1	1	0	1	0	1	0	1	0	0	0	1	…	0	0	0	0	0	0	0	0	0	0	0.073546	0.294118	4	1.743348	2

Figure 3.1: Shortened example of the dataset csv after all the changes.

The categorical feature values were also changed according to the ordinal encoding method that changes the categories with arranged integers expressing the original hierarchy.

Through the testing device it was possible to gather the energy consumption in Watts of these apps over a period of 5 minutes, from this data, the average of energy consumption of each app was calculated. These averages were then split into 3 categories these were:

- Low: For apps whose average consumption is below 1.207 Watts;
- Medium: Apps with a consumption above or equal to 1.207 but below 2.228 Watts fit in this category ;
- High: For all other apps with a consumption above or equal to 2.228 Watts.

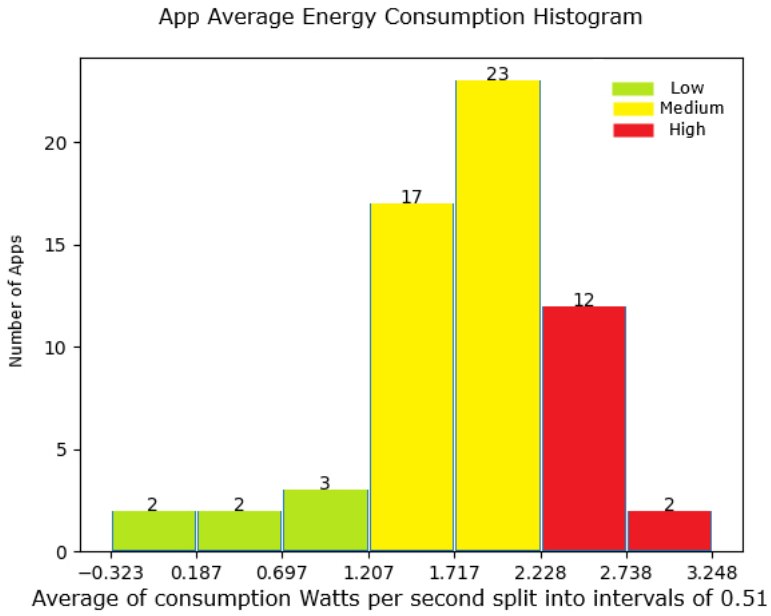

Figure 3.2: Histogram of apps average Watt consumption per second.

The averages and the categorical separation of these made up the target values for the models described further on this chapter, the average was used for the regression models and the categories for the categorical models.

3.2 Test Device

The apps were tested on an Asus Zenfone 4 Max that has been modified so has to function without a battery 3.3 and its connected to an arduino Nano that will register the energy being consumed by the device through a program developed in python that registers the values from each test in a csv file. This device served the purpose of supplying ground truth values when testing the apps from the previously mentioned dataset.

Figure 3.3: Ground Truth setup using Asus Zenfone 4 Max.

The python code also outputs a graph containing the various energy changes gathered during the testing phase of the application 3.4. This allows for a better visualization of how energy consumption shifts whilst the app is being executed by the device.

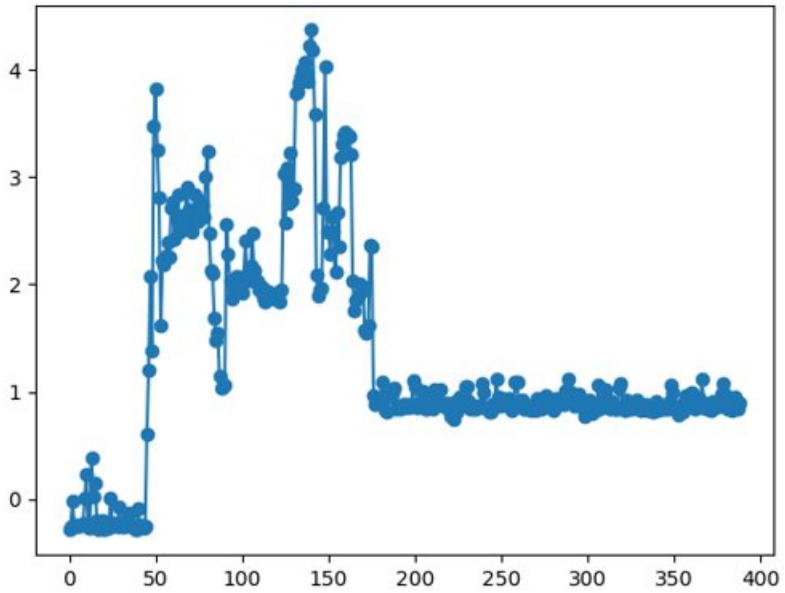

Figure 3.4: Graph containing data points gathered throughout testing in Watts.

3.3 Concepts

3.3.1 Train and Test Set

The concept of train and test set is the existence of a dataset to train the machine learning model and another dataset comprised of never before seen by the model data. These are useful to understand how the model behaves in a real scenario, where new data that it has never processed will occur.
For this project, the train and the test set represent a split in the original dataset mentioned previously.

3.3.2 Prediction

Prediction is the resulting answers to new never before seen data fitted into a previously trained machine learning model.

3.3.3 Feature Selection

Feature selection is a process that occurs before fitting the data to a model, in which the machine is tasked with reducing the amount of features in the dataset so that there's less features for the machine to analyze but more relevant information towards predicting the result accurately.
This project makes use of the method of sequential feature selection. The method focuses on greedily finding which features have the most impact on the expected result and keeping a specific amount of these.

3.3.4 Cross Validation

Cross Validation is a process used to test the quality of a certain model on predicting the outputs of a certain dataset. This is achieved through different methods of splitting the chosen data into train and test sets.
K-Fold and Stratified K-Fold were the chosen methods to perform cross validation of the different models with the dataset.

3.4 Models

For this experiment, 4 different models were tested on the dataset, these were:

- Linear Regression: a linear model that assumes a linear relationship between the inputs and their respective outputs. A good model to solve linear problems. ;
- Support Vector Regression: support vector model which attempts to fit a line within two boundaries creating an hyperplane;

- Naive Bayes: machine learning model that uses the Naive Bayes Theorem to solve categorical classification problems;
- K-Nearest Neighbors: a classification model that predicts results through the proximity that the samples have in the plane.

3.5 Conclusion

Through the development of the dataset, with the help of the app testing setup, it is now possible to apply the machine learning models and the concepts mentioned so to understand if it is possible to either predict the consumption value or its grade from the information gathered from the app's app store.

Chapter 4

Result Analysis and Threats to Validity

This chapter addresses the results obtained from the fitting of the gathered dataset into the previously mentioned models and their performance was evaluated through adequate metrics and the usage of the programming language Python with the scikit-learn library. Two different tests were performed on the dataset using the models. The first test is comprised of splitting the original dataset into a training and testing set, the training set then undergoes the process of feature selection, with varying amounts of resulting features, and cross validation, with varying amounts of folds, before finally attempting to predict the values of the test set. The second test was comprised of passing the dataset through the process of cross validation.
Both regression and classification models were tested using a base feature set, which is comprised of only the permissions each app has totaling to 89 features, and a extra feature set, which adds the minimum android version, total number of permissions and size of the app as features and adds up to 92 features in total.
All splits of the dataset and all shuffling of the dataset that occurs in the cross validation process was given a random state value of 42.

4.1 Regression

4.1.1 Test Conditions

To test the ability of regression for this problem, the following models were used:

- Linear Regression;
- SVR, using 3 different kernels:
  - Linear kernel;
  - Polynomial kernel;
  - Radian Basis Function kernel.

No values were altered in the models utilized, however, the size of the split varies between, 50% for test and training, 80% training and 20% test or 90% training and 10% test. The amount of features changes between all features, 67 for base feature set and 69 for extra feature set representing 75% of the total amount, or 45 for base feature set and 46 for extra feature set corresponing to 50% of all the features.
For Cross Validation the method K-Fold was applied with suffling of the dataset and tested with 3,5 and 10 folds so to better understand the performance of the models on the dataset given.

Metric	Split Size	% of Features	Linear Regression	SVR(Linear)	SVR(Poly)	SVR(RBF)
R^2	50%	All Features	-2.35	-1.87	-0.68	-0.62
		75%	-1.32	-1.23	-0.84	-0.70
		50%	-1.62	-1.41	-0.98	-0.99
	80%	All Features	-8.13	-4.44	-1.48	-1.58
		75%	-4.44	-1.80	-1.45	-1.49
		50%	-1.69	-1.83	-2.05	-1.02
	90%	All Features	-4.70	-2.22	-1.11	-1.19
		75%	-1.29	-1.06	-0.98	-1.39
		50%	$-1.81\times1e24$	-1.15	-1.23	-1.34
MSE	50%	All Features	0.68	0.58	0.34	0.33
		75%	0.47	0.45	0.37	0.34
		50%	0.53	0.49	0.40	0.40
	80%	All Features	2.10	1.25	0.57	0.59
		75%	1.25	0.64	0.56	0.57
		50%	0.62	0.65	0.70	0.47
	90%	All Features	1.97	1.11	0.73	0.76
		75%	0.79	0.71	0.68	0.83
		50%	$6.26\times1e23$	0.74	0.77	0.81

Table 4.1: Prediction Values for the various train test splits with Base Features.

4.1.2 Result Analysis

The results of the regression models are detailed in the following tables 4.1,4.2,4.3,4.4,4.5,4.6, 4.7,4.8,4.9,4.10.
From the results gathered we can affirm that linear regression is a model prone to overfitting this dataset, as seen by the results of 0.41 R^2 and 0.09 MSE with 50% features in 4.3 but scoring -3.02 $\times$ 1e24 R^2 and 1.04$\times$1e24 MSE for a test set of 10% of the original dataset with 50% features.
The support vector regression models present much more optmistic results with RBF presenting the best values in general both in R^2 and MSE.
The difference between the base feature set and extra feature set is also noticeable with the extra features helping in reducing the error of the predictions as seen when comparing 4.1 and 4.2

4.2 Classification

4.2.1 Test Conditions

For classification the models chosen were:

- K-Nearest Neighbors with the values of K:
  - 3;
  - 5;
  - 7;
- Naive Bayes.

Metric	Split Size	% of Features	Linear Regression	SVR(Linear)	SVR(Poly)	SVR(RBF)
R^2	50%	All Features	-6.36	-1.62	-0.79	-0.31
		75%	-1.31	-1.16	-0.94	-0.36
		50%	-1.62	-1.49	-1.05	-0.51
	80%	All Features	-10.17	-4.26	-1.93	-0.90
		75%	-3.44	-2.68	-2.40	-1.29
		50%	-4.20	-2.58	-2.40	-1.22
	90%	All Features	-10.39	-2.66	-1.75	-0.92
		75%	-1.65	-1.04	-1.51	-0.97
		50%	$-3.02\times1e24$	-0.74	-1.67	-0.97
MSE	50%	All Features	1.48	0.53	0.36	0.27
		75%	0.47	0.44	0.39	0.28
		50%	0.53	0.50	0.41	0.31
	80%	All Features	2.57	1.21	0.67	0.44
		75%	1.02	0.85	0.78	0.53
		50%	1.20	0.82	0.78	0.51
	90%	All Features	3.94	1.27	0.95	0.67
		75%	0.92	0.71	0.87	0.68
		50%	$1.04\times1e24$	0.60	0.92	0.68

Table 4.2: Prediction Values for the various train test splits with Extra Features.

Metric	% of Features	Nº of Folds	Linear Regression	SVR(Linear)	SVR(Poly)	SVR(RBF)
R^2	All Features	3	-0.68	-0.46	-0.02	0.10
		5	-1.84	-1.35	-0.42	-0.22
		10	.9.25	-8.54	-6.39	-4.93
	75%	3	-0.20	0.17	0.10	0.34
		5	-0.69	-0.30	-0.40	0.12
		10	-7.42	-5.09	-2.69	-4.09
	50%	3	0.41	0.38	0.21	0.44
		5	0.05	-0.00	-0.03	0.18
		10	-8.09	-2.51	-2.39	-2.72
MSE	All Features	3	0.24	0.21	0.16	0.15
		5	0.32	0.27	0.18	0.16
		10	0.32	0.24	0.17	0.14
	75%	3	0.16	0.13	0.14	0.12
		5	0.20	0.18	0.18	0.13
		10	0.14	0.11	0.13	0.10
	50%	3	0.09	0.11	0.12	0.11
		5	0.18	0.16	0.14	0.13
		10	0.08	0.07	0.11	0.10

Table 4.3: Cross Validation Values for Train Test split of 50% with Base Features

Metric	% of Features	Nº of Folds	Linear Regression	SVR(Linear)	SVR(Poly)	SVR(RBF)
R^2	All Features	3	-2.75	-0.45	0.08	0.14
		5	-4.92	-1.29	-0.45	-0.35
		10	-48.05	-9.96	-7.67	-4.81
	75%	3	-0.25	-0.16	0.24	0.21
		5	-0.73	-0.51	-0.03	-0.06
		10	-7.44	-5.42	-4.47	-4.45
	50%	3	0.41	0.39	0.35	0.37
		5	-0.05	-0.03	0.10	0.08
		10	-8.09	-2.63	-4.89	-3.39
MSE	All Features	3	0.92	0.22	0.15	0.15
		5	0.85	0.26	0.18	0.18
		10	1.66	0.22	0.15	0.16
	75%	3	0.16	0.16	0.13	0.14
		5	0.21	0.20	0.14	0.15
		10	0.14	0.14	0.10	0.13
	50%	3	0.09	0.11	0.11	0.12
		5	0.18	0.16	0.12	0.13
		10	0.08	0.07	0.09	0.10

Table 4.4: Cross Validation Values for Train Test split of 50% with Extra Features

Metric	% of Features	Nº of Folds	Linear Regression	SVR(Linear)	SVR(Poly)	SVR(RBF)
R^2	All Features	3	-1.49	-0.47	0.19	0.24
		5	-2.41	-0.43	0.13	0.17
		10	-4.78	-1.21	-0.17	-0.21
	75%	3	-4.54	0.12	0.15	0.16
		5	-4.28	0.08	0.20	0.11
		10	$-1.34\times1e25$	-0.16	-0.05	-0.16
	50%	3	0.09	0.15	0.27	0.26
		5	0.09	0.12	0.40	0.25
		10	-0.32	0.06	0.09	0.05
MSE	All Features	3	0.42	0.25	0.15	0.15
		5	0.56	0.24	0.16	0.15
		10	0.77	0.29	0.16	0.16
	75%	3	0.95	0.17	0.16	0.16
		5	0.91	0.18	0.15	0.16
		10	$1.71\times1e24$	0.17	0.14	0.15
	50%	3	0.16	0.16	0.14	0.15
		5	0.17	0.17	0.12	0.14
		10	0.19	0.15	0.13	0.13

Table 4.5: Cross Validation Values for Train Test split of 80% with Base Features

Metric	% of Features	Nº of Folds	Linear Regression	SVR(Linear)	SVR(Poly)	SVR(RBF)
R^2	All Features	3	-10506.10	-0.34	0.24	0.21
		5	-7709.25	-0.48	0.18	0.15
		10	-7323.30	-1.82	-0.01	-0.15
	75%	3	-0.98	-0.07	0.35	0.14
		5	-0.37	0.03	0.34	0.10
		10	-0.73	-0.46	0.10	-0.27
	50%	3	-0.30	0.13	0.27	0.25
		5	-0.17	0.29	0.34	0.22
		10	-0.16	-0.06	0.06	-0.05
MSE	All Features	3	1291.60	0.24	0.14	0.15
		5	1708.30	0.27	0.15	0.16
		10	844.98	0.34	0.14	0.16
	75%	3	0.36	0.18	0.12	0.16
		5	0.27	0.18	0.12	0.16
		10	0.24	0.21	0.12	0.17
	50%	3	0.21	0.15	0.13	0.14
		5	0.23	0.14	0.13	0.15
		10	0.16	0.15	0.13	0.15

Table 4.6: Cross Validation Values for Train Test split of 80% with Extra Features

Metric	% of Features	Nº of Folds	Linear Regression	SVR(Linear)	SVR(Poly)	SVR(RBF)
R^2	All Features	3	-2.86	-0.90	-0.09	-0.02
		5	-11.13	-2.75	-0.51	-0.32
		10	-18.99	-3.91	-0.62	-0.33
	75%	3	-0.65	-0.76	-0.04	0.16
		5	-0.79	-0.50	-0.19	0.07
		10	-0.72	-0.51	-0.34	-0.09
	50%	3	-0.27	-0.10	0.03	0.19
		5	-0.50	-0.35	0.04	0.18
		10	-0.31	-0.57	0.06	0.05
MSE	All Features	3	0.68	0.34	0.20	0.18
		5	1.56	0.53	0.23	0.20
		10	1.94	0.51	0.20	0.18
	75%	3	0.27	0.30	0.19	0.15
		5	0.28	0.25	0.19	0.15
		10	0.23	0.22	0.18	0.15
	50%	3	0.22	0.19	0.17	0.14
		5	0.22	0.20	0.16	0.14
		10	0.17	0.19	0.14	0.14

Table 4.7: Cross Validation Values for Train Test split of 90% with Base Features

Metric	% of Features	Nº of Folds	Linear Regression	SVR(Linear)	SVR(Poly)	SVR(RBF)
R^2	All Features	3	-5357.06	-0.83	0.04	0.02
		5	-1942.81	-2.78	-0.31	-0.24
		10	-3544.42	-3.75	-0.25	-0.23
	75%	3	-4.44	-0.40	0.19	0.17
		5	-2.15	-1.36	-0.09	-0.08
		10	-2.27	-1.29	-0.11	-0.11
	50%	3	-0.49	-0.40	0.25	0.24
		5	-1.49	-1.25	-0.05	0.04
		10	-0.66	-1.33	0.03	0.05
MSE	All Features	3	917.60	0.33	0.17	0.17
		5	513.10	0.54	0.21	0.20
		10	1187.02	0.51	0.16	0.17
	75%	3	0.99	0.25	0.14	0.15
		5	0.52	0.39	0.17	0.18
		10	0.49	0.30	0.15	0.17
	50%	3	0.26	0.24	0.13	0.14
		5	0.41	0.35	0.17	0.16
		10	0.23	0.29	0.14	0.14

Table 4.8: Cross Validation Values for Train Test split of 90% with Extra Features

Metric	% of Features	Nº of Folds	Linear Regression	SVR(Linear)	SVR(Poly)	SVR(RBF)
R^2	All Features	3	-31.09	-2.32	-0.75	-0.74
		5	-13.32	-2.46	-0.33	-0.44
		10	-17.81	-2.85	-0.50	-0.49
	75%	3	-1.93	-1.36	-0.47	-0.60
		5	-1.53	-1.30	-0.16	-0.33
		10	$-1.45\times1e21$	-0.97	-0.30	-0.50
	50%	3	$-6.78\times1e28$	-1.27	-0.46	-0.49
		5	-0.62	-1.14	-0.15	-0.30
		10	-0.85	-1.16	-0.29	0.36
MSE	All Features	3	6.13	0.63	0.33	0.33
		5	2.51	0.62	0.27	0.28
		10	2.17	0.61	0.26	0.26
	75%	3	0.57	0.45	0.28	0.31
		5	0.46	0.40	0.22	0.25
		10	$2.52\times1e20$	0.31	0.22	0.24
	50%	3	$1.31\times1e28$	0.43	0.28	0.28
		5	0.30	0.39	0.22	0.24
		10	0.29	0.32	0.22	0.23

Table 4.9: Cross Validation Values for Full Dataset with Base Features

Metric	% of Features	Nº of Folds	Linear Regression	SVR(Linear)	SVR(Poly)	SVR(RBF)
R^2	All Features	3	-5948.73	-2.66	-0.79	-0.47
		5	-8521.00	-2.25	-0.42	-0.23
		10	-35.13	-2.85	-0.37	-0.43
	75%	3	-1.52	-0.51	-0.77	-0.24
		5	-0.79	-0.39	-0.34	-0.08
		10	-0.89	-0.36	-0.41	-0.21
	50%	3	-0.99	-0.30	-0.41	-0.20
		5	-0.46	-0.17	-0.16	-0.08
		10	-0.66	-0.15	-0.21	-0.15
MSE	All Features	3	1162.00	0.69	0.34	0.28
		5	1202.27	0.59	0.28	0.24
		10	5.02	0.62	0.27	0.25
	75%	3	0.48	0.29	0.34	0.24
		5	0.35	0.25	0.26	0.21
		10	0.31	0.24	0.25	0.22
	50%	3	0.38	0.25	0.27	0.23
		5	0.28	0.22	0.22	0.21
		10	0.28	0.20	0.21	0.21

Table 4.10: Cross Validation Values for Full Dataset with Extra Features

The conditions for testing these models are the same as the ones described in 4.1.1

4.2.2 Result Analysis

The results of the classification models are detailed in the following tables 4.11,4.12,4.13,4.14, 4.15,4.16,4.17,4.18,4.19,4.20.
In terms of classification the results show that the dataset doesn't fit the models in an adequate way, with most scores across all metrics being below 0.5 with few exceptions.
The extra feature set doesn't seem to cause a meaningful impact when applied in the classification models with the scores it generated being very close to the scores attained from the base feature set.

4.3 Threats to Validity

This project has a few caveats that may put its quality in question, these are:

- Dataset is very small, for prediction and datascience, adequate datasets usually contain 500+ samples;
- The app testing setup was homemade, therefore the quality of its measurings doesn't compare to the ones made with the appropriate state of the art hardware;
- The predictions are performed on the average consumption value, which ignores possible energy spikes when using certain phone capabilities that are acessed by specific permissions;
- All measurements where extracted from one device, this may provide inconsistency when comparing to other devices since each device has its own energy consumption.

Metric	Split Size	% of Features	KNN(3n)	KNN(5n)	KNN(7n)	Naive Bayes
Accuracy	50%	All Features	0.34	0.48	0.45	0.41
		75%	0.41	0.38	0.48	0.34
		50%	0.45	0.38	0.45	0.48
	80%	All Features	0.33	0.42	0.33	0.42
		75%	0.33	0.33	0.33	0.50
		50%	0.42	0.33	0.25	0.42
	90%	All Features	0.33	0.33	0.33	0.50
		75%	0.33	0.33	0.33	0.17
		50%	0.33	0.33	0.00	0.17
Precision	50%	All Features	0.22	0.29	0.26	0.40
		75%	0.24	0.20	0.29	0.25
		50%	0.26	0.20	0.23	0.49
	80%	All Features	0.21	0.26	0.22	0.30
		75%	0.21	0.23	0.23	0.48
		50%	0.25	0.23	0.19	0.14
	90%	All Features	0.22	0.22	0.28	0.33
		75%	0.22	0.25	0.22	0.06
		50%	0.22	0.25	0.00	0.06
Recall	50%	All Features	0.30	0.43	0.37	0.43
		75%	0.35	0.29	0.40	0.37
		50%	0.37	0.29	0.34	0.37
	80%	All Features	0.31	0.38	0.31	0.42
		75%	0.31	0.31	0.31	0.44
		50%	0.38	0.31	0.24	0.33
	90%	All Features	0.50	0.50	0.50	0.67
		75%	0.50	0.50	0.50	0.33
		50%	0.50	0.50	0.00	0.33
F1-Score	50%	All Features	0.25	0.35	0.31	0.37
		75%	0.28	0.24	0.33	0.28
		50%	0.31	0.24	0.27	0.31
	80%	All Features	0.25	0.31	0.26	0.33
		75%	0.25	0.26	0.26	0.38
		50%	0.30	0.26	0.21	0.20
	90%	All Features	0.30	0.30	0.36	0.44
		75%	0.30	0.30	0.30	0.10
		50%	0.30	0.30	0.00	0.10

Table 4.11: Prediction Values for the various train test splits with Base Features.

Metric	Split Size	% of Features	KNN(3n)	KNN(5n)	KNN(7n)	Naive Bayes
Accuracy	50%	All Features	0.41	0.45	0.45	0.41
		75%	0.41	0.38	0.45	0.41
		50%	0.45	0.38	0.45	0.48
	80%	All Features	0.42	0.42	0.42	0.42
		75%	0.50	0.33	0.42	0.50
		50%	0.42	0.25	0.42	0.50
	90%	All Features	0.33	0.33	0.33	0.50
		75%	0.33	0.33	0.33	0.17
		50%	0.33	0.33	0.17	0.17
Precision	50%	All Features	0.24	0.26	0.27	0.40
		75%	0.24	0.20	0.27	0.29
		50%	0.26	0.20	0.23	0.49
	80%	All Features	0.26	0.25	0.26	0.30
		75%	0.36	0.23	0.26	0.48
		50%	0.25	0.17	0.26	0.48
	90%	All Features	0.25	0.25	0.25	0.33
		75%	0.22	0.22	0.22	0.06
		50%	0.22	0.25	0.08	0.06
Recall	50%	All Features	0.35	0.37	0.37	0.43
		75%	0.35	0.29	0.37	0.44
		50%	0.37	0.29	0.34	0.37
	80%	All Features	0.38	0.38	0.38	0.42
		75%	0.49	0.31	0.38	0.44
		50%	0.38	0.24	0.38	0.44
	90%	All Features	0.50	0.50	0.50	0.67
		75%	0.50	0.50	0.50	0.33
		50%	0.50	0.50	0.33	0.33
F1-Score	50%	All Features	0.28	0.31	0.31	0.37
		75%	0.28	0.24	0.31	0.32
		50%	0.31	0.24	0.27	0.31
	80%	All Features	0.30	0.30	0.30	0.33
		75%	0.41	0.26	0.30	0.38
		50%	0.30	0.20	0.30	0.38
	90%	All Features	0.30	0.30	0.30	0.44
		75%	0.30	0.30	0.30	0.10
		50%	0.30	0.30	0.13	0.10

Table 4.12: Prediction Values for the various train test splits with Extra Features.

Metric	% of Features	N° of Folds	KNN(3n)	KNN(5n)	KNN(7n)	Naive Bayes
Accuracy	All Features	3	0.68	0.68	0.64	0.43
		5	0.66	0.69	0.69	0.47
		10	0.68	0.73	0.73	0.52
	75%	3	0.68	0.64	0.64	0.50
		5	0.66	0.65	0.69	0.44
		10	0.77	0.67	0.73	0.55
	50%	3	0.71	0.64	0.72	0.54
		5	0.66	0.62	0.73	0.51
		10	0.80	0.67	0.65	0.60
Precision	All Features	3	0.42	0.40	0.29	0.27
		5	0.51	0.51	0.50	0.33
		10	0.57	0.60	0.60	0.41
	75%	3	0.42	0.29	0.29	0.37
		5	0.51	0.37	0.50	0.26
		10	0.66	0.51	0.60	0.47
	50%	3	0.54	0.29	0.41	0.29
		5	0.51	0.34	0.58	0.32
		10	0.68	0.51	0.57	0.46
Recall	All Features	3	0.44	0.43	0.39	0.32
		5	0.53	0.56	0.56	0.37
		10	0.63	0.68	0.68	0.47
	75%	3	0.44	0.39	0.39	0.35
		5	0.53	0.46	0.56	0.30
		10	0.74	0.61	0.68	0.49
	50%	3	0.50	0.39	0.48	0.36
		5	0.53	0.42	0.65	0.36
		10	0.77	0.61	0.61	0.55
F1-Score	All Features	3	0.39	0.39	0.33	0.27
		5	0.49	0.50	0.51	0.32
		10	0.57	0.63	0.63	0.42
	75%	3	0.39	0.33	0.33	0.32
		5	0.49	0.39	0.51	0.27
		10	0.68	0.54	0.63	0.46
	50%	3	0.47	0.33	0.44	0.31
		5	0.49	0.37	0.59	0.33
		10	0.71	0.54	0.56	0.48

Table 4.13: Cross Validation Values for Train Test split of 50% with Base Features.

Metric	% of Features	Nº of Folds	KNN(3n)	KNN(5n)	KNN(7n)	Naive Bayes
Accuracy	All Features	3	0.64	0.64	0.57	0.43
		5	0.66	0.62	0.65	0.51
		10	0.73	0.67	0.70	0.52
	75%	3	0.68	0.64	0.64	0.43
		5	0.66	0.65	0.69	0.48
		10	0.77	0.67	0.73	0.48
	50%	3	0.71	0.64	0.72	0.64
		5	0.66	0.62	0.77	0.48
		10	0.80	0.67	0.68	0.63
Precision	All Features	3	0.41	0.31	0.26	0.27
		5	0.51	0.39	0.38	0.36
		10	0.62	0.53	0.53	0.41
	75%	3	0.42	0.29	0.29	0.28
		5	0.51	0.37	0.49	0.28
		10	0.66	0.51	0.60	0.39
	50%	3	0.54	0.29	0.41	0.38
		5	0.51	0.34	0.63	0.25
		10	0.68	0.51	0.59	0.46
Recall	All Features	3	0.43	0.41	0.35	0.32
		5	0.53	0.43	0.46	0.44
		10	0.71	0.61	0.63	0.47
	75%	3	0.44	0.39	0.39	0.31
		5	0.53	0.46	0.54	0.34
		10	0.74	0.61	0.68	0.42
	50%	3	0.50	0.39	0.48	0.41
		5	0.53	0.42	0.68	0.32
		10	0.77	0.61	0.63	0.57
F1-Score	All Features	3	0.37	0.34	0.30	0.27
		5	0.49	0.38	0.40	0.37
		10	0.64	0.56	0.57	0.42
	75%	3	0.39	0.33	0.33	0.28
		5	0.49	0.39	0.50	0.30
		10	0.68	0.54	0.63	0.39
	50%	3	0.47	0.33	0.44	0.37
		5	0.49	0.37	0.64	0.28
		10	0.71	0.54	0.59	0.49

Table 4.14: Cross Validation Values for Train Test split of 50% with Extra Features.

Metric	% of Features	Nº of Folds	KNN(3n)	KNN(5n)	KNN(7n)	Naive Bayes
Accuracy	All Features	3	0.51	0.47	0.62	0.44
		5	0.60	0.49	0.53	0.47
		10	0.59	0.49	0.53	0.42
	75%	3	0.56	0.64	0.64	0.36
		5	0.58	0.67	0.64	0.44
		10	0.63	0.65	0.68	0.36
	50%	3	0.51	0.64	0.67	0.42
		5	0.62	0.67	0.64	0.40
		10	0.62	0.67	0.64	0.52
Precision	All Features	3	0.43	0.28	0.39	0.32
		5	0.49	0.33	0.36	0.38
		10	0.44	0.31	0.28	0.25
	75%	3	0.41	0.60	0.54	0,30
		5	0.43	0.56	0.52	0.29
		10	0.44	0.43	0.47	0.22
	50%	3	0.50	0.61	0.54	0.32
		5	0.50	0.59	0.53	0.25
		10	0.48	0.44	0.44	0.32
Recall	All Features	3	0.41	0.35	0.44	0.39
		5	0.48	0.33	0.36	0.36
		10	0.51	0.37	0.38	0.36
	75%	3	0.45	0.46	0.46	0.33
		5	0.45	0.50	0.48	0.34
		10	0.52	0.50	0.56	0.29
	50%	3	0.48	0.49	0.54	0.39
		5	0.50	0.52	0.55	0.30
		10	0.54	0.53	0.55	0.40
F1-Score	All Features	3	0.39	0.30	0.39	0.34
		5	0.46	0.32	0.33	0.34
		10	0.44	0.31	0.31	0.28
	75%	3	0.42	0.48	0.46	0.30
		5	0.40	0.50	0.47	0.29
		10	0.45	0.45	0.50	0.22
	50%	3	0.45	0.51	0.52	0.34
		5	0.47	0.52	0.48	0.25
		10	0.48	0.47	0.46	0.34

Table 4.15: Cross Validation Values for Train Test split of 80% with Base Features.

Metric	% of Features	Nº of Folds	KNN(3n)	KNN(5n)	KNN(7n)	Naive Bayes
Accuracy	All Features	3	0.51	0.51	0.60	0.44
		5	0.56	0.49	0.60	0.47
		10	0.52	0.54	0.56	0.42
	75%	3	0.51	0.62	0.62	0.42
		5	0.62	0.62	0.64	0.47
		10	0.65	0.68	0.67	0.44
	50%	3	0.51	0.62	0.67	0.51
		5	0.62	0.62	0.62	0.53
		10	0.62	0.65	0.62	0.52
Precision	All Features	3	0.35	0.35	0.40	0.32
		5	0.40	0.32	0.41	0.38
		10	0.35	0.29	0.31	0.25
	75%	3	0.49	0.65	0.48	0.29
		5	0.54	0.55	0.43	0.33
		10	0.55	0.45	0.39	0.25
	50%	3	0.50	0.65	0.59	0.38
		5	0.50	0.55	0.39	0.37
		10	0.48	0.44	0.37	0.27
Recall	All Features	3	0.39	0.39	0.44	0.39
		5	0.41	0.35	0.42	0.36
		10	0.43	0.39	0.40	0.36
	75%	3	0.49	0.51	0.44	0.38
		5	0.56	0.49	0.46	0.34
		10	0.63	0.53	0.51	0.31
	50%	3	0.48	0.51	0.49	0.45
		5	0.50	0.49	0.45	0.36
		10	0.54	0.51	0.48	0.36
F1-Score	All Features	3	0.36	0.36	0.41	0.34
		5	0.39	0.33	0.39	0.34
		10	0.36	0.32	0.34	0.28
	75%	3	0.45	0.50	0.43	0.31
		5	0.52	0.49	0.41	0.31
		10	0.55	0.47	0.43	0.27
	50%	3	0.45	0.50	0.49	0.40
		5	0.47	0.49	0.40	0.34
		10	0.48	0.46	0.40	0.30

Table 4.16: Cross Validation Values for Train Test split of 80% with Extra Features.

Metric	% of Features	N° of Folds	KNN(3n)	KNN(5n)	KNN(7n)	Naive Bayes
Accuracy	All Features	3	0.49	0.45	0.51	0.37
		5	0.47	0.47	0.51	0.49
		10	0.55	0.49	0.53	0.44
	75%	3	0.43	0.47	0.59	0.55
		5	0.53	0.59	0.59	0.51
		10	0.59	0.60	0.61	0.59
	50%	3	0.53	0.59	0.41	0.45
		5	0.62	0.57	0.41	0.63
		10	0.64	0.55	0.33	0.62
Precision	All Features	3	0.38	0.33	0.35	0.31
		5	0.30	0.26	0.24	0.46
		10	0.36	0.29	0.25	0.35
	75%	3	0.27	0.37	0.35	0.20
		5	0.40	0.34	0.39	0.28
		10	0.47	0.31	0.36	0.28
	50%	3	0.59	0.46	0.40	0.23
		5	0.57	0.43	0.34	0.42
		10	0.48	0.35	0.26	0.34
Recall	All Features	3	0.45	0.35	0.36	0.34
		5	0.36	0.30	0.30	0.44
		10	0.43	0.35	0.33	0.40
	75%	3	0.31	0.38	0.38	0.31
		5	0.43	0.42	0.40	0.33
		10	0.50	0.42	0.40	0.38
	50%	3	0.47	0.44	0.43	0.28
		5	0.57	0.41	0.36	0.47
		10	0.52	0.41	0.39	0.44
F1-Score	All Features	3	0.39	0.34	0.34	0.30
		5	0.32	0.28	0.26	0.41
		10	0.38	0.31	0.28	0.34
	75%	3	0.29	0.36	0.33	0.24
		5	0.40	0.37	0.36	0.29
		10	0.46	0.35	0.36	0.31
	50%	3	0.44	0.43	0.33	0.25
		5	0.52	0.39	0.31	0.41
		10	0.49	0.36	0.27	0.37

Table 4.17: Cross Validation Values for Train Test split of 90% with Base Features.

Metric	% of Features	Nº of Folds	KNN(3n)	KNN(5n)	KNN(7n)	Naive Bayes
Accuracy	All Features	3	0.57	0.55	0.49	0.31
		5	0.53	0.55	0.53	0.45
		10	0.51	0.53	0.57	0.42
	75%	3	0.41	0.47	0.55	0.47
		5	0.51	0.61	0.63	0.53
		10	0.57	0.58	0.65	0.55
	50%	3	0.49	0.59	0.41	0.55
		5	0.62	0.57	0.31	0.61
		10	0.64	0.55	0.37	0.62
Precision	All Features	3	0.46	0.40	0.28	0.26
		5	0.32	0.33	0.30	0.42
		10	0.32	0.33	0.31	0.32
	75%	3	0.27	0.37	0.43	0.21
		5	0.37	0.35	0.32	0.27
		10	0.45	0.33	0.38	0.21
	50%	3	0.55	0.46	0.40	0.26
		5	0.57	0.43	0.25	0.38
		10	0.48	0.35	0.33	0.34
Recall	All Features	3	0.46	0.44	0.33	0.25
		5	0.39	0.37	0.37	0.40
		10	0.38	0.40	0.39	0.37
	75%	3	0.30	0.38	0.39	0.32
		5	0.40	0.44	0.40	0.34
		10	0.47	0.43	0.48	0.33
	50%	3	0.45	0.44	0.43	0.34
		5	0.57	0.41	0.30	0.43
		10	0.52	0.41	0.42	0.44
F1-Score	All Features	3	0.43	0.41	0.30	0.25
		5	0.34	0.33	0.32	0.38
		10	0.33	0.35	0.33	0.31
	75%	3	0.28	0.36	0.37	0.24
		5	0.37	0.38	0.34	0.29
		10	0.44	0.37	0.42	0.25
	50%	3	0.41	0.43	0.33	0.28
		5	0.52	0.39	0.21	0.38
		10	0.49	0.36	0.31	0.37

Table 4.18: Cross Validation Values for Train Test split of 90% with Extra Features.

Metric	% of Features	Nº of Folds	KNN(3n)	KNN(5n)	KNN(7n)	Naive Bayes
Accuracy	All Features	3	0.51	0.46	0.56	0.47
		5	0.47	0.45	0.43	0.40
		10	0.49	0.36	0.44	0.43
	75%	3	0.49	0.58	0.61	0.49
		5	0.46	0.56	0.53	0.39
		10	0.56	0.54	0.50	0.51
	50%	3	0.51	0.54	0.53	0.37
		5	0.47	0.56	0.49	0.42
		10	0.61	0.58	0.60	0.52
Precision	All Features	3	0.44	0.37	0.36	0.37
		5	0.41	0.34	0.28	0.32
		10	0.33	0.26	0.27	0.35
	75%	3	0.27	0.49	0.50	0.25
		5	0.23	0.43	0.36	0.18
		10	0.39	0.36	0.35	0.21
	50%	3	0.38	0.33	0.38	0.27
		5	0.30	0.42	0.35	0.32
		10	0.43	0.38	0.38	0.33
Recall	All Features	3	0.43	0.37	0.41	0.43
		5	0.39	0.37	0.32	0.37
		10	0.44	0.33	0.34	0.38
	75%	3	0.36	0.47	0.51	0.34
		5	0.31	0.45	0.43	0.29
		10	0.48	0.46	0.42	0.32
	50%	3	0.38	0.37	0.41	0.33
		5	0.33	0.43	0.39	0.35
		10	0.52	0.44	0.47	0.43
F1-Score	All Features	3	0.39	0.35	0.37	0.37
		5	0.38	0.34	0.29	0.29
		10	0.36	0.28	0.29	0.34
	75%	3	0.30	0.44	0.47	0.28
		5	0.26	0.43	0.39	0.22
		10	0.40	0.39	0.37	0.24
	50%	3	0.34	0.33	0.38	0.25
		5	0.29	0.39	0.35	0.30
		10	0.45	0.40	0.41	0.36

Table 4.19: Cross Validation Values for Full Dataset with Base Features.

Metric	% of Features	Nº of Folds	KNN(3n)	KNN(5n)	KNN(7n)	Naive Bayes
Accuracy	All Features	3	0.51	0.61	0.58	0.44
		5	0.47	0.56	0.51	0.40
		10	0.44	0.56	0.52	0.43
	75%	3	0.61	0.58	0.60	0.47
		5	0.54	0.60	0.58	0.35
		10	0.55	0.57	0.62	0.52
	50%	3	0.51	0.61	0.53	0.49
		5	0.49	0.60	0.49	0.40
		10	0.53	0.60	0.60	0.52
Precision	All Features	3	0.32	0.54	0.42	0.35
		5	0.38	0.44	0.37	0.32
		10	0.29	0.40	0.36	0.35
	75%	3	0.45	0.54	0.47	0.24
		5	0.39	0.51	0.41	0.15
		10	0.44	0.42	0.46	0.24
	50%	3	0.53	0.45	0.38	0.42
		5	0.47	0.47	0.35	0.29
		10	0.39	0.44	0.38	0.31
Recall	All Features	3	0.41	0.51	0.44	0.40
		5	0.38	0.46	0.40	0.37
		10	0.39	0.47	0.42	0.38
	75%	3	0.50	0.49	0.47	0.33
		5	0.46	0.50	0.46	0.26
		10	0.47	0.46	0.53	0.34
	50%	3	0.46	0.51	0.41	0.39
		5	0.42	0.49	0.39	0.31
		10	0.47	0.55	0.47	0.38
F1-Score	All Features	3	0.35	0.50	0.40	0.35
		5	0.34	0.43	0.37	0.29
		10	0.31	0.42	0.38	0.34
	75%	3	0.46	0.49	0.44	0.27
		5	0.41	0.49	0.42	0.18
		10	0.44	0.43	0.47	0.27
	50%	3	0.45	0.47	0.38	0.38
		5	0.42	0.46	0.35	0.25
		10	0.42	0.47	0.41	0.33

Table 4.20: Cross Validation Values for Full Dataset with Extra Features.

4.4 Conclusion

After the analysis of the results, it's understandable that even though the dataset is small, the SVRs still attain interesting values that require further work in order to confirm their validity as models to be used in future cases.

Chapter 5

Conclusions and Future Work

This chapter contains a concise explanation of the future steps that this project allows and the final remarks about this thesis.

5.1 Conclusion

Even though no similar solution to the one developed exists, it was possible to perceive from the related work that all the tools and knowledge required for it were available and despite the results not providing a definitive answer to the predicament, they provide useful clues in what steps to take next to further develop this work and produce a robust solution based on machine learning capable of determining the energy consumption of an application through their information in the app store.

5.2 Future Work

In future developments surrounding this work the following suggestions should be taken into account:

- Expand the dataset by gathering more applications, possibly even testing all applications in the dataset in various devices, so has to understand if there could be patterns that the machine can gather and thus produce a solution that can give accurate results independent of the user's device;
- Further expand the list of models to test on the dataset;
- Study the use of deep learning on the measurement files, to attempt to predict energy consumption and associate energy consumption patterns to the permissions the app requires.

Bibliography

[AGH+15] Raja Wasim Ahmad, Abdullah Gani, Siti Hafizah Ab. Hamid, Feng Xia, and Muhammad Shiraz. A review on mobile application energy profiling: Taxonomy, state-of-the-art, and open research issues. *Journal of Network and Computer Applications*, 58:42–59, 2015. xv, 5, 13, 14

[Alm21] Abdullah Mahmoud Almasri. *Google Play Apps ERM: (Energy Rating Model) Multi-Criteria Evaluation Model to Generate Tentative Energy Ratings for Google Play Store Apps*. PhD thesis, 2021. Available from: `http://hdl.handle.net/10284/9671`. xv, 1, 6, 11

[BGN21] Fares Bouaffar, Olivier Le Goaer, and Adel Noureddine. Powdroid: Energy profiling of android applications. In *2021 36th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW)*, pages 251–254. IEEE, 2021. xiii, 19

[CA19] Luis Cruz and Rui Abreu. Emaas: Energy measurements as a service for mobile applications. In *2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*, pages 101–104. IEEE, 2019. xiii, 17, 18

[CH18] Borle S. Romansky S. Chowdhury, S. and A. Hindle. Greenscaler: training software energy models with automatic test generation. *Empirical Software Engineering*, 24:1649–1692, 2018. xiii, 14, 15, 16

[CPJ+21] Yonghun Choi, Seonghoon Park, Seunghyeok Jeon, Rhan Ha, and Hojung Cha. Optimizing energy consumption of mobile games. *IEEE Transactions on Mobile Computing*, pages 1–1, 2021. 2

[DNPP+17] Dario Di Nucci, Fabio Palomba, Antonio Prota, Annibale Panichella, Andy Zaidman, and Andrea De Lucia. Software-based energy profiling of android apps: Simple, efficient and reliable? In *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 103–114. IEEE, 2017. 16

[HF20] Mohammad Abdul Hadi and Fatemeh Hendijani Fard. Reviewviz: Assisting developers perform empirical study on energy consumption related reviews for mobile applications. In *Proceedings of the IEEE/ACM 7th International Conference on Mobile Software Engineering and Systems*, pages 27–30. ACM, 2020. xiii, 20

[HSK+15] Mohammad Ashraful Hoque, Matti Siekkinen, Kashif Nizam Khan, Yu Xiao, and Sasu Tarkoma. Modeling, profiling, and debugging the energy consumption of mobile devices. *ACM Comput. Surv.*, 48(3):1–40, 2015. xv, 5, 12, 13

[PSM+19] Pijush Kanti Dutta Pramanik, Nilanjan Sinhababu, Bulbul Mukherjee, Sanjeevikumar Padmanaban, Aranyak Maity, Bijoy Kumar Upadhyaya, Jens Bo Holm-Nielsen, and Prasenjit Choudhury. Power consumption analysis, measurement, management, and issues: A state-of-the-art review of smartphone battery and energy usage. *IEEE Access*, 7:182113–182172, 2019. xiii, 5, 6

[Wor21] World Data. Energy consumption in portugal, 2021. Available from: `https://www.worlddata.info/europe/portugal/energy-consumption.php` [cited 18 February 2022]. 1